

Optimal resource allocation in networked control systems using viterbi algorithm

Gökhan Çetin¹, M. Sami Fadali²

¹Department of Electrical Engineering Gümüşhane University, Gümüşhane, Turkey

²Department of Electrical & Biomedical Engineering University of Nevada, Reno, Reno, USA

Article Info

Article history:

Received Jul 22, 2020

Revised Mar 3, 2021

Accepted Apr 7, 2021

Keywords:

Bandwidth allocation
Hidden markov process
Network congestion
Networked control systems
Variable sampling period
Viterbi algorithm

ABSTRACT

This paper presents an optimal bandwidth allocation method for a networked control system (NCS) which includes time-driven sensor, event-driven controller and random channels. A hidden markov model (HMM) with a discretized state space is formulated for the random traffic to predict the network states using a suitable data window. Network bandwidth is allocated based on the predicted traffic state subject to bounds on the deterministic traffic that guarantee acceptable NCS performance and do not exceed hardware limitations. Bandwidth allocation uses l_1 minimization of unmet bandwidth demand. A stability condition is derived for a variable but bounded sampling period interval. Computer simulation results show the effect of varying the number of discrete states for the HMM and the window width on bandwidth allocation. The results compare favorably with a published approach based on fuzzy logic.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Gökhan Çetin

Department of Electrical Engineering

Gümüşhane University

Bağlarbaşı Mah., 29100 Merkez/Gümüşhane Merkez/Gümüşhane, Turkey

Email: gokhancetin@gumushane.edu.tr

1. INTRODUCTION

Maximizing data flow is critical in networked control systems (NCS) with minimal resource capacity where several control loops share a common network [1]-[3]. Limited network capacity can result in packet drop or delay. To eliminate delay or packet drop, which may destabilize or adversely affect the response of the NCS, network resources must be used efficiently [4], [5]. Consequently, optimization of resource allocation is a major issue in NCS [6], [7].

Effective optimization and resource allocation require a good mathematical model of the NCS. In a recent paper [8], Ge *et al.* investigated a Markov model and stochastic control strategy for NCS dynamics. Hidden markov model (HMM) of NCS was first proposed in [9]. The authors proposed a stochastic optimal controller design for an NCS with network-induced delay governed by an underlying Markov chain of unknown probability distribution. Using the Baum-Welch algorithm, they obtained the initial distribution and state-transition probability matrix. They assumed three states for the NCS network load: low, medium, and high. Using a sequence of observations, they identified the Markov model and used it to design a stochastic optimal state feedback controller for the NCS. Nilsson also modeled network states as low, medium and high based on network load with transition between states governed by a Markov Chain [10]. The states were not directly observable and an HMM model was used for the network with the NCS treated as a jump linear system. Based on the HMM, he designed an LQG optimal controller to assure system stability. W. Lim, *et al.* [11], the authors proposed a dynamic bandwidth allocation strategy for orthogonal frequency division

multiple access passive optical networks (OFDMA-PONs). Given the dynamic and stochastic nature of network, they used an HMM to predict traffic states. They defined the traffic states with two variables, the mean and the contrast of the bandwidth request observations. The latter characterizes the degree of traffic variation. Dainotti *et al.* [12] also classified packet-level traffic based on HMM. They used several types of traffic data and estimates of packet size and inter-packet time to evaluate their approach.

Li *et al.* [13] used fuzzy logic to construct a bandwidth manager. They determined the lower and upper bounds for the assignable bandwidth using linear matrix inequalities and the resource constraints. Fuzzy logic provides a flexible feedback mechanism to manage NCS that accounts for an upper bound and a lower bound on the sampling period. The proposed approach saves bandwidth and improves overall control performance as compared to fixed bandwidth allocation.

Although a rich literature on bandwidth allocation exist, to our knowledge the problem of bandwidth allocation for a network shared by NCS and random network traffic was only addressed in an earlier paper by [14]. G. Cetin, *et al.* [14], prediction of future traffic used a conditional poisson model. Although this provided a basis for bandwidth allocation, the results can be significantly improved by improving the quality of traffic prediction.

This paper builds on our earlier work to provide better bandwidth allocation for a network with random channels. Multiple links including random traffic, time-driven traffic, and event-driven traffic sharing a bandwidth is assumed. Network simulation data is used to formulate an HMM model for the network. Unlike earlier work, the model has a large number of discrete states that provide a more accurate network representation. Using the current traffic of the random network, the traffic at the next sampling point is predicted utilizing the HMM. Bandwidth of the network is allocated depending on the resource demands of each individual channel by solving a convex l_1 optimization problem based on the predicted traffic for the random channel and the known deterministic traffic. Although it is assumed that network capacity is time-invariant over the planning horizon of the control system, if the capacity is known a priori, the proposed method is valid if the capacity changes with time.

With limited network capacity, resource demand can exceed capacity. This may cause packet dropout and/or delay. As in [14], l_1 minimization with time-varying sampling period for the time-driven channel is used to allocate bandwidth for the random and event-driven channels optimally. If insufficient bandwidth is allocated for the time driven sensor channel. The sampling period for the time-driven channel is increased to reduce traffic. However, the upper bound on the sampling period should be determined to achieve sufficient stability and performance in the closed-loop NCS system [15], [16]. We derive a condition for the stability of a linear time-invariant plant with a bounded sampling period using the results of [17]. The main contributions of the paper are;

- a. The paper proposes bandwidth allocation method for a network that serves random and deterministic traffic.
- b. The paper provides a new adaptive HMM network model with multiple states that can be used for traffic prediction.
- c. The HMM model is based on a sliding data window to allow for time-varying traffic conditions.
- d. The paper presents a new stability condition for a linear time-invariant system with time-varying sampling period.

The remainder of the paper is organized as follows: section 2 describes the HMM model for random data traffic to predict states using Viterbi algorithm. In section 3, the convex l_1 optimization problem and the dynamic sampling problem are described. Section 4 provides simulation results and discussion. Section 5 is the conclusion.

2. HMM OF NETWORK TRAFFIC

HMM is a statistical Markov models whose states are not directly observable. Instead, an output that is stochastically dependent on the hidden states is directly observable. The observed outputs are related to the HMM states by observation emission probabilities. Transition between states is random and is governed by state transition probabilities which indicate the probability of going one state to another. A states sequence from an initial state to end state can be generated by switching from one state to another based on the transition probabilities. The observable sequence of outputs is related to the state sequence by the emission probability distribution [18].

To obtain a HMM of a network, random network traffic level is allocated to five possible classes, which are “very low(VL)”, “low(L)”, “medium(M)”, “high(H)”, “very high(VH)”. These classes can be determined based on total network capacity and network characteristics. With no loss of generality, the classes are identified with integers and are expressed as $c = \{VL, L, M, H, VH\} = \{1, 2, 3, 4, 5\}$.

Assuming that a NCS shares a common network with N nodes that generate random traffic, let $t_i^r(k)$ pac/sec be the traffic of the i^{th} random channel at time k , $i = 1, 2, \dots, N$.
The traffic conditions are defined as follows;
Channel conditions for i^{th} random nodes:

$$\alpha_{l_i}^c \leq t_i^r(k) < \alpha_{u_i}^c, i = 1, 2, \dots, N \quad (1)$$

where $\alpha_{l_i}^c$ is the lower bound for class c and $\alpha_{u_i}^c$ is the upper bound for class c for the i^{th} random channel.

For five different classes and N random channels, we need 5^N HMM states. For example, consider the case of 3 random channels, i.e. $N = 3$, we use the notation

state 1 = {1 1 1} represents {VL VL VL},

state 125 = {5 5 5} represents {VH VH VH} and so on.

The state-transition matrix, whose entries are transition probabilities from one state to the next, can be determined by using relative frequency if enough data is collected.

The emission matrix determines the probability of network congestion based on HMM states given measured outputs. The emission matrix is classified based on total random traffic as
"no congestion very low traffic (NCVLT)",
"no congestion low traffic (NCLT)",
"critically congested (CC)", "high congestion (HC)",
"very high congestion (VHC)"

"critically congested (CC)", "high congestion (HC)",

"very high congestion (VHC)"

$r = \{NCVLT, NCLT, CC, HC, VHC\} = \{1, 2, 3, 4, 5\}$

Let the random traffic be given by

$$\begin{aligned} B_{tr}(k) &= \sum_{i=1}^N t_i^r(k) \\ B_1 &= \sum_{i=1}^N \alpha_{l_i}^{VL}, B_2 = \sum_{i=1}^N \alpha_{l_i}^L, B_3 = \sum_{i=1}^N \alpha_{l_i}^M, \\ B_4 &= \sum_{i=1}^N \alpha_{u_i}^M, B_5 = \sum_{i=1}^N \alpha_{u_i}^H, B_6 = \sum_i \alpha_{u_i}^{VH} \end{aligned} \quad (2)$$

where B_{tr} is the total random traffic, and $B_i, i = 1, \dots, 6$, are traffic bounds for the five network classes.

$$\begin{aligned} \text{If } B_1 &\leq B_{tr} < B_2 \text{ congestion status } r = 1 \\ \text{If } B_2 &\leq B_{tr} < B_3 \text{ congestion status } r = 2 \\ \text{If } B_3 &\leq B_{tr} < B_4 \text{ congestion status } r = 3 \\ \text{If } B_4 &\leq B_{tr} < B_5 \text{ congestion status } r = 4 \\ \text{If } B_5 &\leq B_{tr} < B_6 \text{ congestion status } r = 5 \end{aligned} \quad (3)$$

Based on this classification, we can define the emission matrix to include the probability of congestion states according to the HMM states. Since we have 125 states and 5 observation states, the size of the E will be 5×125 .

The HMM can be defined as $\lambda_n = (N, M, T, E, \pi)$ where T is the state-transition matrix, E is the emission matrix, N is the number of the states of HMM (traffic conditions), M is the number of observation states (congestion conditions) and n is the window size [9]. The HMM model is shown in Figure 1. In the figure, we have;

$q_k \in \theta = [\theta_i]$ is the hidden state of the Markov chain at time $k = 1, \dots, n$, θ is the hidden state, $\theta_i = i, i = 1, 2, \dots, N$
$O_k \in O = [o_i]$ is the observation at time k , $o_i = i, i = 1, 2, \dots, M$
$\pi = [\pi_1, \pi_2, \dots, \pi_N]$ are the initial probabilities of the states where $\pi_i = P(q_k = \theta_i)$ where $i = 1, \dots, N$ and $k = 1, \dots, n$.
$T(q_k, q_{k+1}) = [a_{ij}]$ is the state-transition matrix with $a_{ij} = P(q_{k+1} = \theta_j q_k = \theta_i), 1 \leq i, j \leq N$
$E = [e_{ij}]$ is the emission matrix with $e_{ij} = P(O_k = o_j q_k = \theta_i), 1 \leq i \leq N, 1 \leq j \leq M$

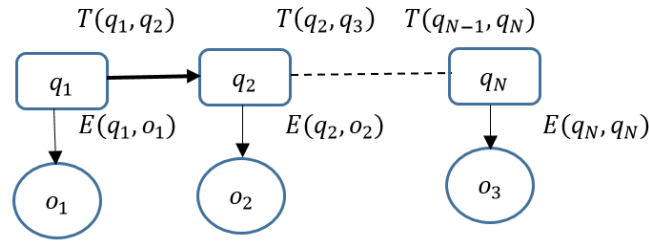


Figure 1. HMM model [10]

To find T for window size n , we first determine the traffic classes from the available data using (1). Then we find the transitions probabilities using the relative frequency of state transitions. To find the emission matrix E for window size n , we use the ranges of (3).

Since network dynamics change over time, we determine the adaptive transition matrix and the emission probability matrix based on random network traffic data over a specified time window. Using n observation o_1, o_2, \dots, o_n , of network traffic conditions data ending at the current time k , we find the adaptive transition matrix and the emission probability matrix for window size n . The Viterbi algorithm [18-22] is used to predict the most likely states at time $k = 1, \dots, n + 1$, we save the predicted state at $k = n + 1$. Then we shift the window by one step, and update the adaptive transition and the emission probability matrices. The Viterbi algorithm predicts the most likely state at time $k = n + 1, \dots, 2n + 1$. We save the predicted state at $k = 2n + 1$ and the process is repeated to predict future states.

The Viterbi algorithm works as follows. Given the HMM model and a sequence of observations, we to estimate the most likely sequence of states [18]. The most likely sequence ending with $q_k = \theta$ is obtained using the utility function

$$v_{\theta}^k = \max_{q_k = \theta} \log P(O_k, q_k) \quad (4)$$

The paths that lead to each state from the possible prior states are explored to find the path that has the highest probability. For each possible new state, only the highest probability path is saved using $V_{\theta'}^{k+1}$.

$$V_{\theta'}^{k+1} = \arg \max_{\theta} v_{\theta}^k + \log (P_{q_{k+1}|q_k}(\theta'|\theta)) + \log (P_{O_{k+1}|q_{k+1}}(O_{k+1}|\theta')) \quad (5)$$

where θ' is posterior state at time $k + 1$, and $V_{\theta'}^{k+1}$ is the maximum cumulative log-probability achieved for $(k + 1)^{th}$ time step from k^{th} time step.

Using (5) recursively, we determine the most likely paths from $k = 1$ to $k = n + 1$ for window size n and save the most likely transitions for backtracking iteration. The final posterior probability of the all sequence of states can be maximized by recursively maximizing the joint probability for each possible new state. The best final state at time $k = n + 1$ for window size n can be calculated as

$$\hat{\theta}^{n+1} = \arg \max_{\theta} V_{\theta}^{n+1} \quad (6)$$

Later backtracking iteration is used to find all hidden states in the window (traffic conditions)

$$\hat{\theta}^{k+1} = V_{\hat{\theta}^{k+1}}^{k+1}, \quad k = n, n - 1, \dots, 1 \quad (7)$$

3. BANDWIDTH ALLOCATION

Figure 2 shows general structure of NCS. Random nodes occupy bandwidth of the network based on its network traffic, sensor, controller and actuator nodes are interconnected via network for data transmission as in shown in Figure 2. Assume n observations are known initially. Traffic conditions c for each random channel at time $k + 1$ are predicted utilizing the HMM model and the Viterbi algorithm. With knowledge

of c , we can assign random traffic for each random channel at time $k + 1$ by assigning the upper bound $\alpha_{u_i}^c$ related to the traffic condition to guarantee the required bandwidth allocation for the time-driven sensor node.

Since the total network bandwidth capacity is constrained, bandwidth is allocated optimally depending on the traffic of each channels. The Viterbi algorithm predicts the random traffic at time $k + 1$ given the traffic of the event-driven channels. Bandwidth is allocated for the random and event-driven channel based on the following optimization as in [14].

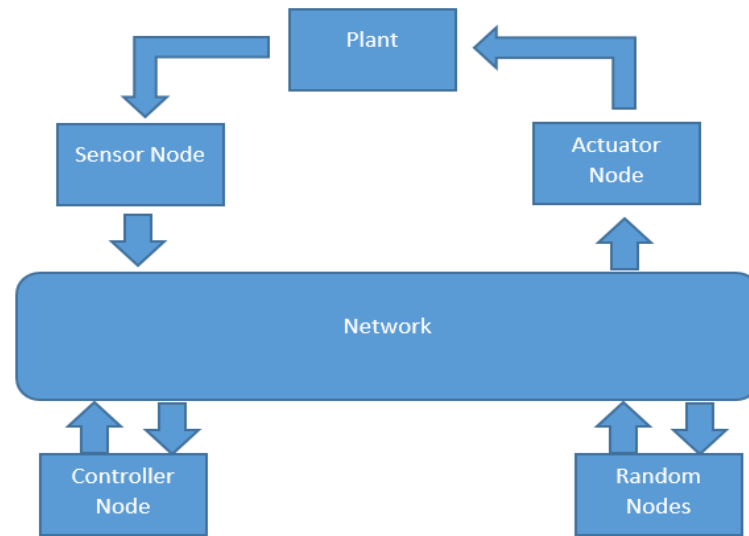


Figure 2. NCS structure [14]

The sampling period of the NCS is varied to optimize bandwidth allocation. Even for a linear time-invariant plant, the closed-loop NCS is not time-invariant and the placement of its eigenvalues inside the unit circle is no longer sufficient for stability. Zhou and Zhao provided the following result regarding the stability of the perturbed discrete linear time-varying system with state matrix $A(k)$ and perturbation matrix $\Delta A(k)$ [17].

$$x(k+1) = (A(k) + \Delta A(k))x(k), \quad k = 0, 1, 2, \dots \quad (8)$$

where $A(k), \Delta A(k) \in \mathcal{R}^{n \times n}$

Theorem 1 [17]:

Assume that the nominal discrete linear time-varying system with state matrix $A(k)$ is uniformly exponentially stable. If there exist a positive constant β and a sufficiently small constant ϵ such that $\Delta A(k)$ satisfies

$$\sum_{j=0}^{k-1} \|\Delta A(j)\| \leq \epsilon k + \beta, \quad k = 1, 2, \dots, \quad (9)$$

then, the perturbed discrete linear time-varying system (8) is uniformly exponentially stable.

Based on the above theorem, we obtain a condition for the stability of a linear time-invariant NCS with time-varying sampling period. We assume that the state matrix has geometric multiplicity equal to its algebraic multiplicity.

Theorem 2 : If the linear time-invariant system with digital control is closed-loop stable with state feedback control gain K and sampling period for the time driven channel of the related control loop h^t , then the system is uniformly exponentially stable with time-varying sampling period in a bounded interval $[h_{NCSmin}^t, h_{NCSmax}^t]$ including h^t .

Proof:

The discrete-time state space model is given by

$$x(k+1) = [e^{Ah^t(k)} - B(h^t(k))]x(k) \quad (10)$$

The state-transition matrix and the input matrix can be written as

$$\begin{aligned} e^{Ah^t(k)} &= \sum_{i=1}^n Z_i e^{\lambda_i h^t(k)} \\ B(h^t(k)) &= \frac{\sum_{i=1}^n Z_i B [e^{\lambda_i h^t(k)} - 1]}{\lambda_i} \end{aligned} \quad (11)$$

For a zero eigenvalue, we have the term $Z_i B h^t(k)$, which can be handled similarly to what we show below. Perturbation in the sampling period will result in the perturbation matrix

$$\Delta A(k) = \sum_{i=1}^n \left(Z_i - \frac{BK}{\lambda_i} \right) [e^{\lambda_i h^t} - e^{\lambda_i h^t(k)}] \quad (12)$$

For a sampling period varying in a bounded interval $[T_{min}, T_{max}]$, we have

$$|e^{\lambda_i h^t} - e^{\lambda_i h^t(k)}| \leq |e^{\lambda_i h_{NCSmax}^t} - e^{\lambda_i h_{NCSmin}^t}| \leq k_1, i = 1, \dots, n \quad (13)$$

Because the nominal dynamics are stable

$$\left\| Z_i - \frac{BK}{\lambda_i} \right\| \leq k_2, i = 1, \dots, n$$

The condition of Theorem-1 [17] becomes

$$\sum_{j=0}^{k-1} \|\Delta A(j)\| \leq \epsilon k, \epsilon = nk_1 k_2 \quad (14)$$

4. RESULTS AND DISCUSSIONS

A DC motor position control example adapted from [23, 24] is simulated to compare our results with fixed bandwidth allocation. The DC motor transfer function is;

$$G(s) = \frac{1000}{s(s+1)}$$

A feedback controller is designed to achieve desired transient response. An upper bound for the sampling period of the time-driven channels $h_{iNCSmax}$, is necessary for obtaining satisfactory NCS performance and is calculated based on the system dynamics. The desired response of the system must have percentage overshoot less than 20% and settling time less than 0.2 s. For 20% overshoot, required damping ratio is approximately 0.46. The natural frequency of the closed loop system is calculated based on the specifications as $\omega_n = 48.73 \text{ rad/s}$. The undamped natural frequency should be slower than the packet arrival to the controller, to obtain the required sampling frequency for the NCS, ω_r is selected as $\omega_r = \alpha \omega$, where α a scale factor selected to reduce aliasing. Assuming $\tau = 1 \text{ s}$, we selected $\alpha = 15$ to obtain $h_{iNCSmax} = \frac{2\pi}{\alpha \omega_d} \cong 11 \text{ ms}$. since τ for the network is 0.1 s, $h_{iNCSmax} = 1.1 \text{ ms}$. Based on hardware limitation on sampling rate, we select $h_{iNCSmin} = 0.6 \text{ ms}$.

For the simulation, frequency-division multiple access (FDMA) network channel is used in NCS as in [25]. The FDMA network channel has a capacity of 495 Mbps and packet size is 100 bytes. Thus, the total network capacity is 495 pac/0.1s. The traffic in the three random channels is predicted and the sensor and the controller channels generate 100pac/0.1s to satisfy required transients response.

Random networks occupy 80% of the bandwidth of the network while time-driven sensor and event-driven controller nodes occupy the remaining 20%. Fixed sampling period of the time-driven sensor channel is chosen as 1ms. If the allocated bandwidth for random channels is high, the allocated bandwidth for the controller and the sensor channels may become insufficient and this may cause delay or package drop which deteriorates the performance of the NCS as shown in Figure 3.

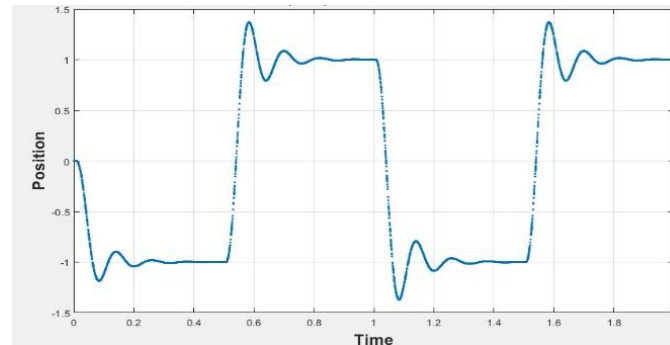


Figure 3. NCS output without bandwidth allocation

Assume that, three random nodes which generate random traffic and two deterministic nodes which are the time-driven sensor node and event-driven controller node are sharing the network in Figure 2. The HMM model proposed in Section 2 with 125 states for three random channels with the channel conditions $\{VL, L, M, H, VH\}$. An adaptive (HMM-1) is constructed by selecting an appropriate window size $n = 10$ and the Viterbi algorithm predicts future traffic volume for random channels. Since rest of the channels are deterministic and they have fixed network parameters, their traffic can be calculated from the parameters. The controller contains an event-driven task and every instances, a sample arrives over the network from the sensor node. The network traffic model used in the paper is offline, however since HMM model of the network traffics are modelled, if priory network traffic in the channels are known in the selected window size this model will work for online network traffic model also.

To evaluate the effect of the number of traffic classes of the HMM model, we construct an HMM model with 27 states for three random channels with three conditions $\{L, M, H\}$, dubbed (HMM-2). The latter is used to predict the next random traffic volume for comparison to HMM-1.

The traffic in the controller to actuator channel is 100 pac/0.1sec. The controller computes a control signal after receiving the packet, then the packet that includes control signal is sent to the event-driven actuator node. After predicting the traffic of the random channel at time $k + 1$, we have values for all network traffic that is used for bandwidth allocation. The bandwidth allocation is $t(k + 1) = [\tilde{t}^r(k + 1), t^t, 100]$, where t^t is to be calculated using the convex optimization of (8).

After allocating bandwidth optimally, the sampling period for the time-driven channel is obtained by employing (9). The sampling period of the sensor to controller channel $h_i^t(k + 1)$ is calculated using (10). For a satisfactory output response, the sampling period of the time-driven channel is adjusted to eliminate channel congestion.

At first, fixed network parameters without bandwidth allocation and sampling rate adjustment is used. Network congestion occurs as the traffic of some channels exceeds their capacity. Figure 3 shows that the closed-loop system response is oscillatory because network congestion causes packet loss and time delay.

The available bandwidth is allocated depending on the demand of each channel. Bandwidth allocation for the random and the event-driven controller channel is achieved using l_1 optimization. Because the total capacity of Network is limited, insufficient bandwidth can be allocated for the time-driven sensor channel and network congestion may occur for the time-driven sensor channel. Its sampling rate is adjusted to satisfy its allocated bandwidth capacity subject to stability and performance constraints.

Figure 4 indicates that the sampling period of the sensor channel stays in the interval $[0, 0.25]$ for both HMM-1 and HMM-2. The sampling period is assigned according to the allocated bandwidth for the time-driven sensor channel. The sampling period never exceeds the prescribed upper and lower bounds $[0.6, 1.1]$ ms. Figure 5 and Figure 6 show that the bandwidth allocation using both HMM-1 and HMM-2 provides satisfactory responses. However, Figure 6 shows that the response of HMM-2 includes sharp perturbations because of the large prediction error and limited sampling period interval. Figure 7 shows the

global network scheduling for the channels in NCS in the Truetime network simulator [24]. The network scheduling varies based on the dynamic sampling period assignment and bandwidth allocation.

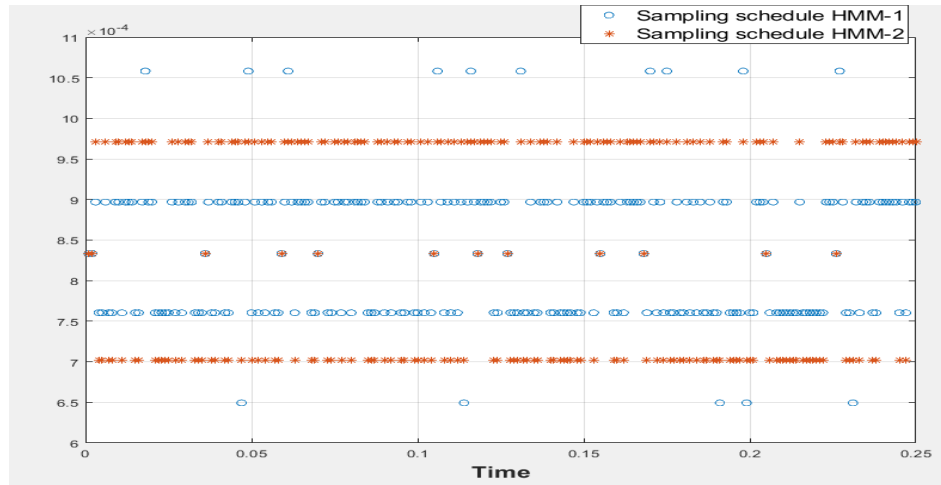


Figure 4. Sampling period of the sensor channel

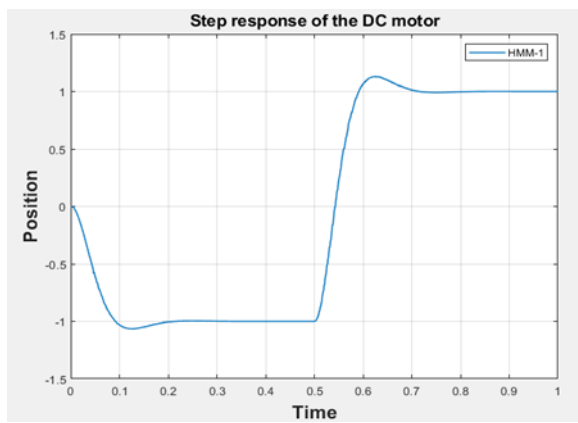


Figure 5. Plant output in NCS with bandwidth allocation (HMM-1)

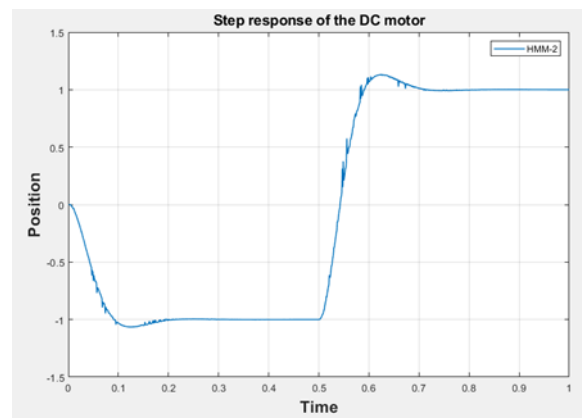


Figure 6. Plant output in NCS with bandwidth allocation (HMM-2)

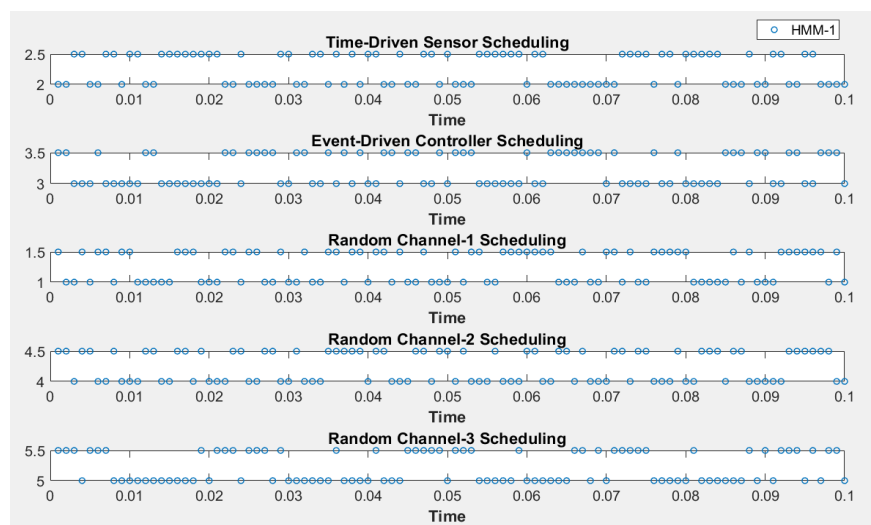


Figure 7. Channels scheduling for HMM-1 (high=running data, low=idle)

Figure 8 indicates bandwidth allocation percentage for each channel according to the network traffic in the three random channels in the time interval $[0, 0.05]$ for both HMM-1 and HMM-2. Based on the predicted states, Figure 9 verifies that HMM-1 provides satisfactory bandwidth allocation and meets traffic demand better than HMM-2.

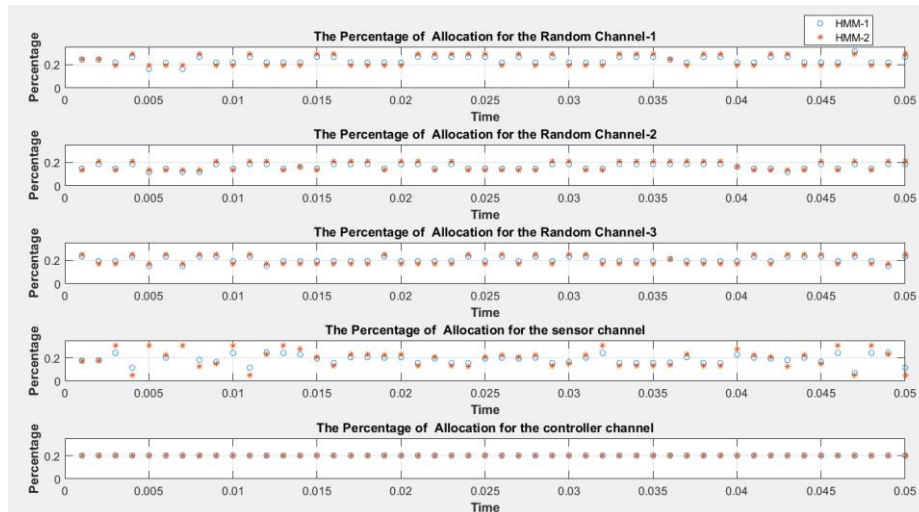


Figure 8. Percentage bandwidth allocation

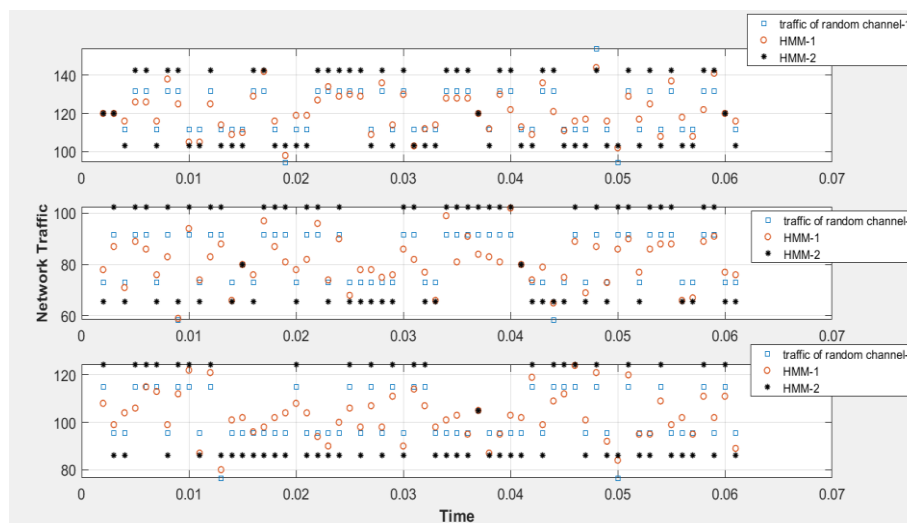


Figure 9. Traffic prediction in random channel

Figures 10, 11, 12 and Table 1 show the predicted random traffic for HMM-1 and HMM-2, respectively. The random traffic in the network is predicted by employing HMM and the Viterbi algorithm and used for bandwidth allocation. The figures show that, prediction errors for the states are lower for HMM-1. This provides more efficient bandwidth allocation and sampling period scheduling based on the allocation of bandwidth for the sensor channel.

Figure 13 and Table 1 reveal that an adequate but small window size provides better state prediction. An excessive window size results in larger state prediction errors because of changes in network dynamics over time that make older data inapplicable. Increasing the number of network states that define the traffic condition reduces the prediction error.

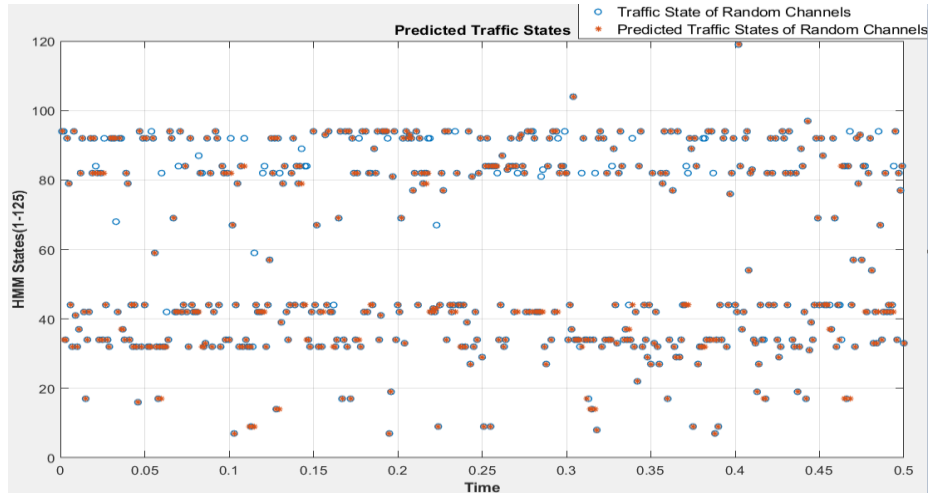


Figure 10. Traffic state prediction in random channel (HMM-1)

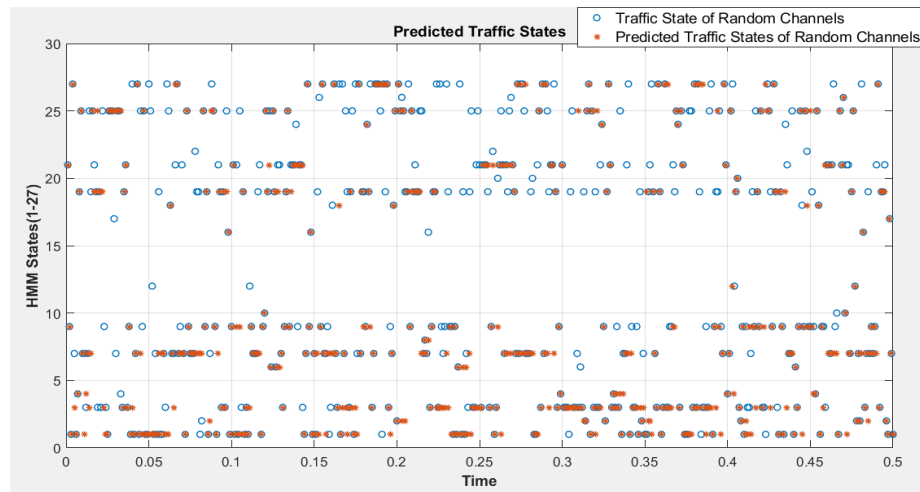


Figure 11. Traffic state prediction in random channel (HMM-2)

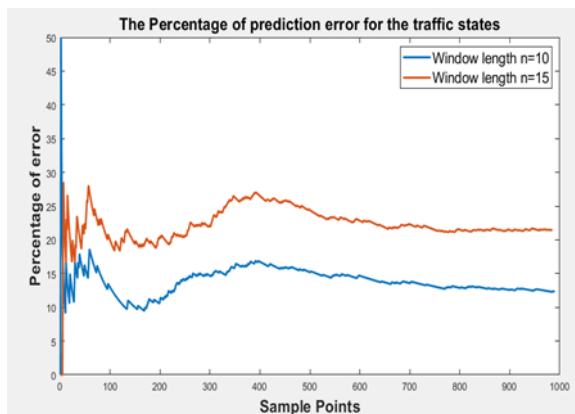
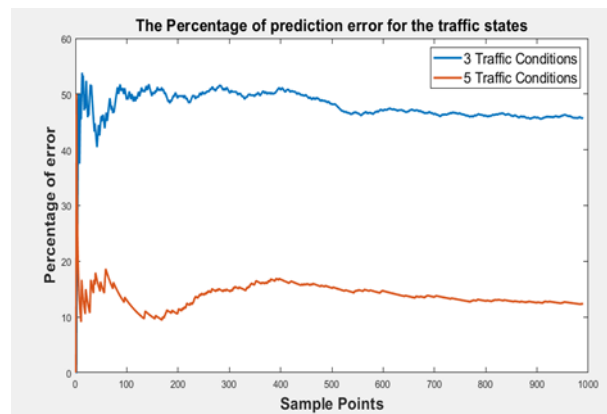
Figure 12. Comparison of prediction error for HMM-1 and HMM-2 with window size $n=10$ Figure 13. Comparison of prediction for window size $n=10$ and window size $n=15$ for HMM-1

Table 1. Comparison of prediction error for HMM-1 and HMM-2 for different window sizes

Window size(n)	5 traffic conditions HMM-1	3 traffic conditions HMM-2
10	12.3232%	45.1066%
15	21.5228%	59.6954%
20	30.0000%	68.3673%
25	38.2564%	73.7436%
30	45.9794%	76.1856%
60	76.4894%	83.5106%
70	81.2603%	83.9785%
80	82.3021%	83.2609%

5. CONCLUSION

This paper provides a new approach for bandwidth allocation for NCS. An adaptive HMM with 5 traffic classes is constructed for random traffic and the traffic is estimated utilizing the Viterbi algorithm. The HMM parameters are updated based on the traffic data for a window of specified width to account for changes in traffic conditions. l_1 Optimization method is employed to allocate bandwidth for the time-driven sensor channel in NCS. The sampling period of the time-driven sensor channels is adjusted to mitigate network traffic and avoid packet loss and delay but kept within bounds dictated by performance requirements and bandwidth limitations. We present conditions for the stability of the resulting time-varying system. A second HMM network with only 3 traffic classes, as is common in the literature, is also designed for comparison to the 5-class HMM. Simulation results where the 5-class HMM outperforms the 3-class HMM demonstrate the importance of having a sufficient number of traffic classes for the HMM. Results for different data window widths demonstrate the importance of selecting an appropriate window width to capture the current traffic dynamics of the network. Future work will apply the methodology to more complex NCS and exploit knowledge of random traffic statistics for bandwidth allocation.

REFERENCES

- [1] R. A. Gupta and M. Chow, "Networked Control System: Overview and Research Trends," in *IEEE Transactions on Industrial Electronics*, vol. 57, no. 7, pp. 2527-2535, July 2010, doi: 10.1109/TIE.2009.2035462.
- [2] T. C. Yang, "Networked Control Systems: a brief survey," *IEE Proceedings-Control Theory and Applications*, vol. 153, no. 4, 10 July 2006.
- [3] Dan Zhang *et al.*, "Analysis and synthesis of networked control systems: A survey of recent advances and challenges," *ISA Transactions*, Vol. 66, pp. 376-392, 2017.
- [4] A. T. Al-Hammouri, M. S. Branicky, V. Liberatore and S. M. Phillips, "Decentralized and dynamic bandwidth allocation in networked control systems," *Proceedings 20th IEEE International Parallel & Distributed Processing Symposium*, Rhodes, Greece, 2006, pp. 8 pp.-, doi: 10.1109/IPDPS.2006.1639398.
- [5] W. Zhiwen, S. Hongtao, "Control and Optimization of Network in Networked Control System," *Mathematical Problems in Engineering*, vol. 2014, article id:237372, 2014.
- [6] W. P. M. H. Heemels, A. R. Teel, N. van de Wouw and D. Nešić, "Networked Control Systems With Communication Constraints: Tradeoffs Between Transmission Intervals, Delays and Performance," in *IEEE Transactions on Automatic Control*, vol. 55, no. 8, pp. 1781-1796, Aug. 2010, doi: 10.1109/TAC.2010.2042352.
- [7] X. Zhang *et al.*, "Networked control systems: a survey of trends and techniques," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 1, pp. 1-17, January 2020, doi: 10.1109/JAS.2019.1911651.
- [8] Y. Ge, *et al.*, "Modeling of Random Delays in Networked Control Systems," *Hindawi Publishing Corporation Journal of Control Science and Engineering*, vol. 2013, article id 383415, <http://dx.doi.org/10.1155/2013/383415>.
- [9] Fu-Chun Liu and Yu Yao, "Modeling and analysis of networked control systems using hidden Markov models," *2005 International Conference on Machine Learning and Cybernetics*, Guangzhou, China, 2005, pp. 928-931 Vol. 2, doi: 10.1109/ICMLC.2005.1527076.
- [10] J. Nilsson, "Real-Time Control Systems with Delays," Department of Automatic Control, Lund Institute of Technology, ISSN 0280-5316, Sweden, 1998.
- [11] W. Lim *et al.*, "Dynamic Bandwidth Allocation for OFDMA-PONs Using Hidden Markov Model," in *IEEE Access*, vol. 5, pp. 21016-21019, 2017, doi: 10.1109/ACCESS.2017.2657549.
- [12] A. Dainotti, W. de Donato, A. Pescapé and P. Salvo Rossi, "Classification of Network Traffic via Packet-Level Hidden Markov Models," *IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference*, New Orleans, LA, USA, 2008, pp. 1-5, doi: 10.1109/GLOCOM.2008.ECP.412.
- [13] Z Li, W Wang, Y Jiang, "Managing Quality-of-Control and Requirement-of-Bandwidth in Networked Control Systems via Fuzzy Bandwidth Scheduling," *International Journal of Control, Automation, and Systems*, vol. 7, no. 2, pp. 89-296, 2009, doi 10.1007/s12555-009-0215-7.
- [14] G. Cetin, M. S. Fadali, H. Xu, "Optimal Resource Allocation in Networked Control Systems," *IET Cyber-Physical Systems: Theory & Applications*, vol. 5, issue. 2, pp. 168-175, 2020.

- [15] M. H. Roohi and T. Chen, "Optimal Sampling Rate and Quantization for Networked Control Systems*," *2019 IEEE 28th International Symposium on Industrial Electronics (ISIE)*, Vancouver, BC, Canada, 2019, pp. 623-628, doi: 10.1109/ISIE.2019.8781450.
- [16] L. Jie, Z. Ying and F. Chunxia, "Stability Analysis for Variable Sampling Networked Control Systems with Data Packet Dropout and Partly Unknown Transition Probabilities," *2019 Chinese Control And Decision Conference (CCDC)*, Nanchang, China, 2019, pp. 2367-2372, doi: 10.1109/CCDC.2019.8832969.
- [17] B. Zhou and T. Zhao, "On Asymptotic Stability of Discrete-Time Linear Time-Varying Systems," in *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 4274-4281, Aug. 2017, doi: 10.1109/TAC.2017.2689499.
- [18] A. M. Fraser, "Hidden Markov Models and Dynamical Systems," Society for Industrial and Applied Mathematics (SIAM), 2008
- [19] G. D. Forney, "The viterbi algorithm," in *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268-278, March 1973, doi: 10.1109/PROC.1973.9030.
- [20] H. -. Lou, "Implementing the Viterbi algorithm," in *IEEE Signal Processing Magazine*, vol. 12, no. 5, pp. 42-52, Sept. 1995, doi: 10.1109/79.410439.
- [21] G. W. Slade, "The Viterbi algorithm demystified," March 2013.
- [22] H. Zulkarnaen and V. Puturuhu, "Viterbi Extraction tutorial with Hidden Markov Toolkit," *ArXiv* abs/1908.03143, 2019.
- [23] A. Cervin, *et al.*, "TrueTime 2.0-Reference Manual," Department of Automatic Control, Lund University, Sweden, April 2016.
- [24] K. -. Arzen, A. Cervin, J. Eker and L. Sha, "An introduction to control and scheduling co-design," *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No.00CH37187)*, Sydney, NSW, Australia, 2000, pp. 4865-4870 vol.5, doi: 10.1109/CDC.2001.914701.
- [25] L. Xie and X. Zhang, "TDMA and FDMA Based Resource Allocations for Quality of Service Provisioning Over Wireless Relay Networks," *2007 IEEE Wireless Communications and Networking Conference*, Hong Kong, China, 2007, pp. 3153-3157, doi: 10.1109/WCNC.2007.582.

BIOGRAPHIES OF AUTHORS



Gökhan Çetin earned a BS in Electrical Engineering from Ankara University in 2012, and a Ph. D. from the University of Nevada, Reno in 2019. He is currently working at Gumushane University as a teaching assistant. My research interests are in the areas of robotics and networked control systems.



M. Sami Fadali earned a BS in Electrical Engineering from Cairo University in 1974, an MS from the Control Systems Center, UMIST, England, in 1977 and a Ph. D. from the University of Wyoming in 1980. He was an Assistant Professor of Electrical Engineering at the University of King Abdul Aziz in Jeddah, Saudi Arabia 1981-1983. From 1983-85, he was a Post Doctoral Fellow at Colorado State University. In 1985, he joined the Electrical Engineering Dept. at the University of Nevada, Reno, where he is currently Professor of Electrical Engineering. In 1994 he was a visiting professor at Oakland University and GM Research and Development Labs. He spent the summer of 2000 as a Senior Engineer at TRW, San Bernardino. His research interests are in the areas of fuzzy logic stability and control, state estimation and fault detection, and applications to power systems, renewable energy, and physiological systems.